

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Canceled)
2. (Currently Amended) A computer-implemented method of generating computer code for a web application, comprising:
 - a computing system receiving an input file files, wherein the input file comprising markup language text for a files are at least one web application graphical user interface;
 - the computing system generating an application framework code and an event handler skeleton, wherein generating an event handler skeleton comprises:
 - the computing system parsing the markup language text at least one input file;
 - the computing system identifying an input tag in the markup language text;
 - the computing system identifying in the markup language text, an attribute within the markup language tag;
 - the computing system identifying in the markup language text a value corresponding to the identified attribute;
 - reviewing the parsed input file for one or more of a tag type, an attribute name and an attribute value; and
 - the computing system using the identified markup language input tag, the identified attribute, and the identified value to identify a rule for creating determining an event handler method based on one or more of the tag type, the attribute name and the attribute value; and
 - the computing system using the identified rule and the identified markup language input tag, the identified attribute, and the identified value to create an event handler method;
 - the computing system receiving web application business logic objects;
 - the computing system receiving event handler methods;

the computing system organizing the application framework code, the web application business logic objects and the event handler methods into application source code; and
the computing system binding the web application source code with the input files at runtime.

3. (Canceled)
4. (Original) The method of claim 2, wherein the web application source code is generated in an object-oriented programming language.
5. (Original) The method of claim 4, wherein the object-oriented programming language is Java.
6. (Original) The method of claim 4, wherein the object-oriented programming language is C++.
7. (Original) The method of claim 2, further comprising determining if the application framework code is available for the web application.
8. (Original) The method of claim 2, further comprising generating a business logic foundation code.
9. (Original) The method of claim 2, further comprising generating a graphical user interface code.
10. (Original) The method of claim 9, wherein generating a graphical user interface code is based on the input files.
11. (Original) The method of claim 2, wherein generating an event handler skeleton is based on the input files.
12. (Original) The method of claim 2, further comprising compiling the web application source code.

13. (Original) The method of claim 2, further comprising interpreting the web application source code.

14. (Original) The method of claim 2, wherein the input files are in XML format.

15. (Original) The method of claim 2, wherein the input files are in HTML format.

16. (Original) The method of claim 2, wherein the input files are in cHTML format.

17. (Original) The method of claim 2, wherein the input files are in WML format.

18. (Original) The method of claim 2, further comprising receiving modified input files.

19. (Original) The method of claim 18, further comprising compiling the modified input files at runtime.

20. (Original) The method of claim 19, further comprising binding the web application source code with the compiled modified input files at runtime.

21. (Original) The method of claim 20, wherein the modified input files are compiled into DOM objects at runtime.

22. (Original) The method of claim 18, further comprising interpreting the modified input files at runtime.

23. (Original) The method of claim 22, further comprising binding the web application source code with the interpreted modified input files at runtime.

24. (Original) The method of claim 2, further comprising generating application runtime properties.

25. (Original) The method of claim 2, further comprising generating application SQL statements.

26. (Original) The method of claim 2, wherein the application framework code comprises an application object and a sen/let web application framework object.

27. (Currently Amended) A computer-implemented method of generating computer code for a web application, comprising:

a computing system receiving an input file files, wherein the input file comprising markup language text for a files are at least one web application graphical user interface;
the computing system retrieving an application framework code from an application directory;
the computing system generating an event handler skeleton, the generating comprising:
the computing system parsing the markup language text;
the computing system identifying an input tag in the markup language text;
the computing system identifying in the markup language text, an attribute within the markup language tag;
the computing system identifying in the markup language text a value corresponding to the identified attribute;
the computing system using the identified markup language input tag, the identified attribute, and the identified value to identify a rule for creating an event handler method; and
the computing system using the identified rule and the identified markup language input tag, the identified attribute, and the identified value to create an event handler method;
receiving web application business logic objects;
receiving event handler methods;
organizing the application framework, code, the web application, business logic objects and the event handler methods into application source code; and
binding the web application source code with the input files at runtime ~~and wherein~~
~~generating an event handler skeleton further comprises:~~
~~parsing at least one input file;~~

~~reviewing the parsed input file for one or more of a tag type, an attribute name and an attribute value; and determining an event handler method based on the tag type, the attribute name and the attribute value.~~

28. (Original) The method of claim 27, further comprising retrieving a business logic foundation code.

29. (Original) The method of claim 27, further comprising generating a business logic foundation code.

30. (Canceled)

31. (Original) The method of claim 27, wherein the web application source code is generated in an object-oriented programming language.

32. (Original) The method of claim 27, further comprising determining if the application framework code is available for the web application.

33. (Original) The method of claim 27, further comprising generating a graphical user interface code.

34. (Original) The method of claim 33, wherein generating a graphical user interface code is based on the input files.

35. (Original) The method of claim 27, wherein generating an event handler skeleton is based on the input files.

36. (Original) The method of claim 27, further comprising compiling the web application source code.

37. (Original) The method of claim 27, further comprising interpreting the web application source code.

38. (Original) The method of claim 27, wherein the input files are in XML format.
39. (Original) The method of claim 27, wherein the input files are in HTML format.
40. (Original) The method of claim 27, wherein the input files are in cHTML format.
41. (Original) The method of claim 27, wherein the input files are in WML format.
42. (Original) The method of claim 27, further comprising receiving modified input files.
43. (Original) The method of claim 42, further comprising compiling the modified input files at runtime.
44. (Original) The method of claim 43, further comprising binding the web application source code with the compiled modified input files at runtime.
45. (Original) The method of claim 42, further comprising interpreting the modified input files at runtime.
46. (Original) The method of claim 45, further comprising binding the web application source code with the interpreted modified input files at runtime.
47. (Original) The method of claim 27, wherein the application framework code comprises an application object and a servlet web application framework object.
48. (Currently Amended) A computer-implemented method of generating computer code for a web application, comprising:
 a computing system receiving an input file files, wherein the input file comprising markup language text for a files are at least one web application graphical user interface;
 the computing system generating an application framework code and an event handler skeleton, generating an event handler skeleton comprising: [[:]]
 the computing system parsing the markup language text;
 the computing system identifying an input tag in the markup language text;

the computing system identifying in the markup language text, an attribute within the markup language tag;

the computing system identifying in the markup language text a value corresponding to the identified attribute;

the computing system using the identified markup language input tag, the identified attribute, and the identified value to identify a rule for creating an event handler method; and

the computing system using the identified rule and the identified markup language input tag, the identified attribute, and the identified value to create an event handler method;

the computing system receiving web application business logic objects;

the computing system receiving event handler methods;

the computing system organizing the application framework code, the web application business logic objects and the event handler methods into web application source code;

the computing system receiving modified input files; and

the computing system binding the modified input files with the web application source code at runtime ~~and wherein generating an event handler skeleton further comprises:~~

~~parsing at least one input file;~~

~~reviewing the parsed input file for one or more of a tag type, an attribute name and an attribute value; and~~

~~determining an event handler method based on the one or more of tag type, the attribute name and the attribute value.~~

49. (Original) The method of claim 48, further comprising compiling the modified input files at runtime.

50. (Original) The method of claim 48, further comprising interpreting the modified input files at runtime.

51. (Canceled)

52. (Original) The method of claim 48, wherein the web application source code is generated in an object-oriented programming language.

53. (Original) The method of claim 48, further comprising determining if the application framework code is available for the web application.

54. (Original) The method of claim 48, further comprising generating a business logic foundation code.

55. (Original) The method of claim 48, further comprising generating a graphical user interface code.

56. (Original) The method of claim 48, further comprising compiling the web application source code.

57. (Original) The method of claim 48, further comprising interpreting the web application source code.

58. (Original) The method of claim 48, wherein the input files are in XML format.

59. (Original) The method of claim 48, wherein the input files are in HTML format.

60. (Original) The method of claim 48, wherein the input files are in cHTML format.

61. (Original) The method of claim 48, wherein the input files are in WML format.

62. (Original) The method of claim 49, wherein the modified input files are compiled into DOM objects at runtime.

63. (Original) The method of claim 48, wherein the application framework code comprises an application object and a servlet web application framework object.

64. (Currently Amended) A computer-implemented method of generating computer code for a web application, comprising:

a computing system receiving an input file files, wherein the input file comprising markup language text for a files are at least one web application graphical user interface;

the computing system retrieving an application framework code from an application directory;

the computing system generating an event handler skeleton, the generating comprising: [[:]]

the computing system parsing the markup language text;

the computing system identifying an input tag in the markup language text;

the computing system identifying in the markup language text, an attribute within the markup language tag;

the computing system identifying in the markup language text a value corresponding to the identified attribute;

the computing system using the identified markup language input tag, the identified attribute, and the identified value to identify a rule for creating an event handler method; and

the computing system using the identified rule and the identified markup language input tag, the identified attribute, and the identified value to create an event handler method;

the computing system receiving web application business logic objects;

the computing system receiving event handler methods;

the computing system organizing the application framework code, the web application business logic objects and the event handler methods into web application source code;
and

the computing system binding modified input files with the web application source code at runtime and wherein generating an event handler skeleton further comprises:

parsing at least one input file;

~~reviewing the parsed input file for a tag type, an attribute name and an attribute value; and
determining an event handler method based on the tag type, the attribute name and the attribute
value.~~

65. (Canceled)

66. (Original) The method of claim 64, further comprising determining if the application framework code is available for the web application.

67. (Original) The method of claim 64, further comprising generating a business logic foundation code.

68. (Original) The method of claim 64, further comprising retrieving a business logic foundation code.

69. (Original) The method of claim 64, further comprising generating a graphical user interface code.

70. (Original) The method of claim 64, wherein generating an event handler skeleton is based on the input files.

71. (Original) The method of claim 64, wherein the input files are in XML format.

72. (Original) The method of claim 64, wherein the input files are in HTML format.

73. (Original) The method of claim 64, wherein the input files are in HTML format.

74. (Original) The method of claim 64, wherein the input files are in WML format.

75. (Original) The method of claim 64, further comprising compiling the modified input files at runtime.

76. (Original) The method of claim 64, further comprising interpreting the modified input files

at runtime.

77. (Original) The method of claim 64, wherein the application framework code comprises an application object and a servlet web application framework object.

78. (Currently Amended) A computer-implemented method of generating computer code for a web application, comprising:

the computing system generating an event handler skeleton wherein the generating comprises:

parsing at least one input file, the input file comprising markup language text for a graphical user interface,

identifying an input tag in the markup language text,

identifying in the markup language text, an attribute within the markup language tag;

identifying in the markup language text a value corresponding to the identified attribute;

using the identified markup language input tag, the identified attribute, and the identified value to identify a rule for creating an event handler method, and

using the identified rule and the identified markup language input tag, the identified attribute, and the identified value to create an event handler method;

~~reviewing the parsed input file for one or more of a tag type, an attribute name and an attribute value, and~~

~~determining an event handler method based on the one or more of tag type, the attribute name and the attribute value;~~

the computing system receiving a business logic foundation code, the event handler skeleton and a graphical user interface code; and

the computing system preparing web application business logic objects based on the business logic foundation code.

79.-161. (Canceled)

162. (Currently Amended) A device for generating computer code for a web application, comprising:

a storage device; and

a processor connected to the storage device, the storage device storing a program for controlling the processor;

the processor operative with the program to:

generate a business logic foundation code, an event handler skeleton, and a graphical user interface code, the event handler skeleton generated by:

parsing markup language text comprised in an input file for a graphical user interface;

identifying an input tag in the markup language text;

identifying in the markup language text, an attribute within the markup language tag;

identifying in the markup language text a value corresponding to the identified attribute;

using the identified markup language input tag, the identified attribute, and the identified value to identify a rule for creating an event handler method; and

using the identified rule and the identified markup language input tag, the identified attribute, and the identified value to create an event handler method;

receive web application business logic objects from a web developer;

receive event handler methods from the web developer;

organize the application framework code, the web application business logic objects and the event handler methods into web application source code;

compile the web application source code;

compile the modified input files at runtime; and

bind the compiled modified input files with the compiled web application source code at runtime ~~and wherein the processor is operative with the program in generating the event handler skeleton to:~~

~~parse at least one input file;~~
~~review the parsed input file for one or more of a tag type, an attribute name and an attribute value; and~~
~~determine an event handler method based on the one or more of a tag type, the attribute name and the attribute value.~~

163. (Currently Amended) A device for generating computer code for a web application, comprising:

a storage device; and
a processor connected to the storage device, the storage device storing a program for controlling the processor;

the processor operative with the program to:

~~receive an input file files, wherein the input file comprising markup language text for a files are at least one web application graphical user interface;~~

~~generate an application framework code and an event handler skeleton, generating an event handler skeleton comprising:~~

~~parsing the markup language text;~~

~~identifying an input tag in the markup language text;~~

~~identifying in the markup language text, an attribute within the markup language tag;~~

~~identifying in the markup language text a value corresponding to the identified attribute;~~

~~using the identified markup language input tag, the identified attribute, and the identified value to identify a rule for creating an event handler method; and~~

~~using the identified rule and the identified markup language input tag, the identified attribute, and the identified value to create an event handler method;~~

receive web application business logic objects; receive event handler methods;

organize the application framework code, the web application business logic objects and the event handler methods into application source code; and

~~bind the web application source code with the input files at runtime and wherein the processor is operative with the program in generating the event handler skeleton to:~~
~~parse at least one input file;~~
~~review the parsed input file for one or more of a tag type, an attribute name and an attribute value;~~
~~and~~
~~determine an event handler method based on the one or more of a tag type, the attribute name and the attribute value.~~

164. (Currently Amended) A device for generating computer code for a web application, comprising:

a storage device; and
a processor connected to the storage device, the storage device storing a program for controlling the processor;
the processor operative with the program to:
receive an input file files, wherein the input file comprising markup language text for a files
~~are at least one web application graphical user interface;~~
retrieve an application framework code from an application directory;
generate an event handler skeleton, the generating comprising:
parsing the markup language text;
identifying an input tag in the markup language text;
identifying in the markup language text, an attribute within the markup language tag;
identifying in the markup language text a value corresponding to the identified attribute;
using the identified markup language input tag, the identified attribute, and the identified value to identify a rule for creating an event handler method;
and

using the identified rule and the identified markup language input tag, the identified attribute, and the identified value to create an event handler method;
receive web application business logic objects;
receive event handler methods;
organize the application framework code, the web application business logic objects and the event handler methods into application source code; and
bind the web application source code with the input files at runtime ~~and wherein the processor is operative with the program in generating the event handler skeleton to:~~
~~parse at least one input file;~~
~~review the parsed input file for one or more of a tag type, an attribute name and an attribute value; and~~
~~determine an event handler method based on the one or more of a tag type, the attribute name and the attribute value.~~

165. (Currently Amended) A device for generating computer code for a web application, comprising:
a storage device; and
a processor connected to the storage device, the storage device storing a program for controlling the processor;
the processor operative with the program to:
receive an input file files, wherein the input files file comprising markup language text for a
~~are at least one web application graphical user interface;~~
generate ~~an application framework code and an event handler skeleton,~~ generating an event handler skeleton comprising:
parsing the markup language text;
identifying an input tag in the markup language text;
identifying in the markup language text, an attribute within the markup language tag;

identifying in the markup language text a value corresponding to the identified attribute;

using the identified markup language input tag, the identified attribute, and the identified value to identify a rule for creating an event handler method;
and

using the identified rule and the identified markup language input tag, the identified attribute, and the identified value to create an event handler method;

receive web application business logic objects; receive event handler methods;

organize the application framework code, the web application business logic objects and the event handler methods into web application source code;

receive modified input files; and

bind the modified input files with the web application source code at runtime ~~and wherein the processor is operative with the program in generating the event handler skeleton to:~~

~~parse at least one input file;~~

~~review the parsed input file for one or more of a tag type, an attribute name and an attribute value; and~~

~~determine an event handler method based on the one or more of a tag type, the attribute name and the attribute value.~~

166. (Currently Amended) A device for generating computer code for a web application, comprising:

a storage device; and

a processor connected to the storage device, the storage device storing a program for controlling the processor;

the processor operative with the program to:

receive an input file files, wherein the input file comprising markup language text for a files
~~are at least one web application graphical user interface;~~

retrieve an application framework code from an application directory;

generate an event handler skeleton, generating an event handler skeleton comprising:

parsing the markup language text;
identifying an input tag in the markup language text;
identifying in the markup language text, an attribute within the markup
language tag;
identifying in the markup language text a value corresponding to the
identified attribute;

using the identified markup language input tag, the identified attribute, and
the identified value to identify a rule for creating an event handler method;
and

using the identified rule and the identified markup language input tag, the
identified attribute, and the identified value to create an event handler method;

receive web application business logic objects;

receive event handler methods;

organize the application framework code, the web application business logic objects and the
event handler methods into web application source code;

receive modified input files; and

bind the modified input files with the web application source code at runtime ~~and wherein the~~
~~processor is operative with the program in generating the event handler skeleton to:~~

~~parse at least one input file;~~

~~review the parsed input file for one or more of a tag type, an attribute name and an attribute~~
~~value; and~~

~~determine an event handler method based on the one or more of a tag type, the attribute name~~
~~and the attribute value.~~

167. (Previously Presented) The method of claim 2, further comprising:

determining if an application framework code is available for the web application; and if the
application framework code is not available, then generating the application framework code.

168. (Previously Presented) The device of claim 166, further comprising:

a determining mechanism configured to determine if an application framework code is available for the web application; and
a code generator configured to generate the application framework code.

169. (Previously Presented) The device of claim 162, wherein the processor is further operative with the program to:

determine if an application framework code is available for the web application; and if the application framework code is not available, then generate the application framework code.

170. (Canceled)

171. (Canceled)

172. (Canceled)

173. (Canceled)

174. (Canceled)